CERTIFICATE OF ELECTRONIC TRANSMISSION
I hereby certify that this correspondence is being electronically transmitted to the United States Patent and Trademark Office on 01 June 2006.
/Kathryn Marley/
Kathryn Marley

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

John O. Lamping et al.

Application No. **09/124,805**

Confirmation No. **7115**

Filed: **29 July 1998**

Title: **Local Relative Layout of Node-Link Structures in Space with Negative Curvature**

Group Art Unit: 2628

Examiner: Jin Cheng Wang

CUSTOMER NO. 22470

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL REPLY BRIEF

Sir:

This Appeal Reply Brief is filed in support of Appellants' appeal from the Final Office Action, mailed 04 November 2005 in this case, and in response to the Examiner's Answer mailed 03 April 2006.

Should it be determined that any fees are required in connection with this communication, the Commissioner is hereby authorized to charge those fees to Deposit Account No. 50-0869 (Attorney Docket No. INXT 1002-1).

///

TABLE OF CONTENTS

I. AGREED POINTS

The Examiner's Answer acknowledges Appellants' statement identifying the real party in interest, the lack of related appeals or interferences, the status of amendments after final, the summary of claimed subject matter, and the grounds of rejection to be reviewed on appeal. It also acknowledges the copy of appealed claims in the appendix and takes no issue with the evidentiary appendix.

Regarding the substantive positions taken in the Examiner's Answer, the Examiner appears now to agree that an arc, as shown in some of the *display space* drawings in Lamping '632, does not constitute a "space with negative curvature" as called for in some of the claims on appeal. In particular, the Examiner's Answer, in the Grounds of Rejection, has deleted all arguments that rely on an arc constituting a "space with negative curvature".

The Examiner also appears to agree now that the methods taught in Lamping for mapping node-link elements from a space with negative curvature (layout space) into flat display space, are not relevant to Appellants' claims since citations to those sections of Lamping no longer appear in the Examiner's Answer. For the first time, the Answer now cites sections of Lamping that describe his methods for *laying out* a node-link structure *into* the space with negative curvature. However, as explained below, those teachings also fail to anticipate Appellants' claims.

Additionally, it appears that the Examiner now agrees that dependent claims 31-36 should be patentable, since he does not appear to have responded to the points raised by Appellants in the Appeal Brief with respect to claims 32-36, and appears to have responded to Appellants' points regarding claim 31 only perfunctorily.

II. STATUS OF CLAIMS

The Examiner's Answer acknowledges that the statement of the claims contained in the appeal brief is correct.

However, concurrently with this Appeal Reply Brief, Appellants are filing an AMENDMENT UNDER 37 C.F.R. 41.33(b)(1) to cancel claims 17-28.

Thus assuming the Amendment is entered, and Appellants see no reason why it should not be entered, **only claims 29-44** remain in present application and remain subject to the present appeal.

III. REPLY TO EXAMINER'S ARGUMENTS

In Section (10) of the Examiner's Answer, Response to Argument, the Examiner sets forth the Examiner's synopsis of Appellants' arguments, and then answers them.

The arguments set forth with respect to claims 17-28 are considered moot since these claims have now been canceled.

Appellants therefore will now reply to each of the Examiner's arguments regarding claims 29-44. Appellants respectfully submit that the rejection of these claims under 35 U.S.C. 102(b) as being anticipated by Lamping et al. should be reversed.

A. Independent Claim 29, as Representative of Independent Claims 29 and 42-44 and Dependent Claims 30 and 37-41

In the Appeal Brief, Appellants pointed out that all of the Examiner's rejections of these claims relied on parts of Lamping '632 that describe a method for mapping from *layout* space to a circular *display* region, which is flat, and is not a space with negative curvature as called for in claim 29.

The Examiner appears now to have accepted this deficiency and has made new arguments based on the parts of Lamping '632 that describe a method for *laying out* a node-link structure *into layout space* (a space with negative curvature).

Respectfully, however, Appellants are having difficulty understanding the Examiner's new argument. Appellants therefore submit that having withdrawn the previous arguments for unpatentability, the Examiner has now failed to make a *prima facie* case in the new argument.

Nevertheless, Appellants will point out some of the reasons why even Lamping's "layout" teachings cannot anticipate claim 29, and then, in case the Board discerns a point in the Examiner's argument not answered by the discussion below, Appellants will attempt to answer arguments that the Examiner *might* be making.

1. Detailed Analysis of the Lamping Patents and their Relationship to Claim 29

The following analysis is based on pages 11-20 of Appellants' Response D, filed October 21, 2003, made in response to an argument that Appellants had thought the Examiner made orally during the telephone interview on July 2, 2003. Unfortunately the analysis is necessarily detailed; Appellants are certain that the Examiner has not analyzed Lamping in equivalent detail.

Claim 29 calls for, among other things, a step of:

> "storing the positions for each element [of a plurality of elements] in a data structure such that <u>after the positions for all elements in the plurality</u> have been calculated, <u>the position of each element in the plurality</u> is stored in the data structure <u>only relative</u> to an element of the node-link structure other than a root element of the node-link structure." (emphasis added).

Thus since a plurality means at least two, this claim requires at a minimum that the positions of *at least two* elements of the node-link structure, other than a root element, be stored in the data structure only in relative terms.

By storing the positions of the elements in layout space only in relative terms, a change in the node-link structure does not require traversal of the entire node-link structure, from the root node outward, in order to reflect the change in the layout space.

It will be seen below that Lamping '632 does not teach that the positions of *any* elements of the node-link structure be stored in the data structure only in relative terms, and certainly does not teach that the positions of *at least two* be stored only in relative terms.

a. <u>Lamping '632 does not itself specify whether he stores positions of elements of the node-link structure be stored in the data structure in relative terms</u>

The sections of Lamping '632 that discuss the step of obtaining "layout data" in the space with negative curvature are set forth primarily in Fig. 9 and the accompanying text at col. 20, line 18 through col. 21, line 8. The routine operates recursively, starting with the handle of a root node in the node-link data, with "coordinates of a position" for the root node in the hyperbolic plane, and with a pie-shaped region (wedge and room bound) within which the node's children will be laid out. (Lamping '632, col. 20, lines 20-28.) Thus before any children of the root node are laid out, the "position" of the root node is already known.

If the node currently being handled has children, then the routine calculates a "distance" from the current node to its children (step 354). The distance is one coordinate for each of the child nodes, but Lamping '632 does not indicate that a second coordinate, such as an angle, is known at this point for any of the children. It will be seen below that a related case, Lamping '250, incorporated by reference into Lamping '632 for purposes of detailing the layout operations, reveals that the second coordinate is *not* known at this point. Thus at this point in the routine, the "position" is not yet *fully* known, either in relative or absolute terms, for any of the children of the root node.

The routine then begins an iterative loop that handles each of the current node's children in turn. Each iteration begins (in step 362) by obtaining, for the *next* child node, a "position in the hyperbolic plane", and a wedge and room bound within which the child node's further children will be laid out. Thus by the end of step 362, *all* coordinates of the *current* child's "position" are known. In addition, it can be seen that the "position" of each child is known before the routine loops around to begin handling the next sibling.

In step 364, the routine then makes a recursive call to lay out the current child's *further* children (if any), providing the handle of the current child and its "position", wedge, and room bound from step 362. Thus before the call is made to lay out any *further* children (step 364), the "position" of the current child and the *distance* to its further children are known. Again, however, Lamping '632 does not indicate that the *angle* to any of the further children are known at this point. That determination is left for the performance of step 364 in the context of handling each individual further child node.

Whenever the recursion reaches the depth of a leaf node (a node with no further children, step 352), the routine determines a "radius" for the node (step 370), and creates a "data structure" for the node (step 372). The routine creates such a data structure also for each parent node after data structures have already been created for all the children of the parent node (step 360). As pointed out at Lamping '632, col. 21, lines 2-3, the data structure for the node includes its "position", and its "radius" from step 370). (Note that the "radius" of a node is not part of its "position"; it is only an indication of how much space will be allocated to the node for text and other visual features upon subsequent rendering onto a display. See Lamping '632, col. 24, lines 18-32.)

Nothing in any of the above excerpts from Lamping '632 states whether the node "positions" identified in these data structures are expressed in absolute or relative terms after

completion of the layout routine. While one or more coordinates of the position may have been expressed in relative terms at some point as an *intermediate step* during the calculation, Lamping '632 does *not* teach that any node position is expressed in relative terms *as written into the final data structure*.

b. Lamping '250 specifies that the positions are stored in absolute terms

Nor does *Lamping '250*, referred to in Lamping '632 as the "Layout Application", set forth any example in which a node position is expressed in relative terms as written into the final data structure. Lamping '632 invokes Lamping '250 to describe a technique for obtaining a "distance" to the children in the hyperbolic plane, in step 354 (Lamping '632, col. 20, line 43). But step 354 is only an intermediate step in the developing of the position of the current node, to be stored in the final data layout structure. The distance to the children is used to *determine* each child node's "position", wedge and room bound (Lamping '632, col. 20, lines 46-50). The Lamping '250 example does not say that this distance is ever represented in the final Layout data structure.

Lamping '632 invokes Lamping '250 also to describe techniques for performing such determination of each child node's "position", wedge and room bound based on the "distance" (Lamping '632, col. 20, lines 49-50). One might therefore look to Lamping '250 for any teaching that the "position" calculated by such techniques is relative or absolute. But Lamping '250, too, never teaches that any "position" calculated for a *final* data structure, be expressed in relative terms.

In Lamping '250, the layout process example described therein is described primarily with respect to Figs. 9 and 10 and the text at col. 21,line 33 - col. 24, line 46. Fig. 9 and the accompanying description at '250, col. 21, line 33 - col. 22, line 40, is very similar to the discussion in Lamping '632 of the step of obtaining layout data in the space with negative curvature. Lamping '250 then goes on, however, to discuss details of certain steps of his Fig. 9, including step 362, "Obtain Next Child's Position, Wedge, Room Bound", for a particular implementation.

The discussion refers to Fig. 10, which includes a step 400 of "...Obtain Child's Position, Wedge, Room Available". As it relates the obtaining of the child's "position", Lamping '250 begins his explanation of this step as follows:

```
The act in box 400 can obtain preliminary width angle
Wn as the product of the child's weight and the angle
```

> fraction, both obtained in box 390. The act in box 400 can use (θx, θy) and the previous width angle W0, either from box 398 or from the previous iteration, to calculate child direction (θx, θy) having coordinate values equal to (((θx (cos (W0 +Wn)))-(θy (sin (W0 +Wn)))), ((θy (cos (W0 +Wn)))+(θx (sin (W0 +Wn))))). Then the act in box 400 updates previous width W0 to have the value of the preliminary width Wn. (Lamping '250, col. 23, lines 26-34; emphasis added).

Thus the first step in calculating the child's "position" is to calculate its second coordinate, the *angle*. Clearly, therefore, the "position" of the current node in layout space was not fully known, in relative or absolute terms, prior to step 400 (i.e. prior to step 362 in Lamping '632).

Lamping '250 then continues:

> The act in box 400 can obtain the next child node's position using the position (x, y) of the parent node, from box 350 if the parent node is the root node or from box 364 when performed for the parent node; using the child direction (θx, θy); and using the distance D from box 394 or 396. The position include values (x, y, θx, θy) that are calculated using transformations. (Lamping '250, col. 23, lines 35-40; emphasis added).
>
> The act in box 400 can obtain a first translator from the origin to the position of the parent node, then obtain the new x- and y-coordinates by applying the first translator to the x- and y-components of the distance D, (Dθx, Dθy), where (θx, θy) is the child direction described above. The act in box 400 can then obtain a second translator to the origin for the new x- and y- coordinates, and apply the second translator to the results of applying the first translator to the child direction (θx, θy), to obtain the new values for ((θx, θy). (Lamping '250, col. 24, lines 4-10; emphasis added).

In other words, the "position" (x- and y- coordinates) of the current child node position in layout space is obtained by starting with the (x,y) "position" of the parent node in layout space, and then applying the distance D and direction (θx, θy) from the parent node to the child node, to obtain new x- and y- coordinates in layout space for the child node. This is a transformation which *converts* a relative position (Dθx, Dθy), which establishes the child node's position only *relative* to the parent node's position, into an absolute position, which establishes the child node's position relative to the *origin*.

Thus by the end of step 400 in Lamping '250, the current node position is expressed in *absolute* terms, not *relative* to any other node position (except for children of the root node, for which the absolute and relative positions may be the same).

And this is the expression of the node position that is passed back to step 364 at the bottom of the flow chart of Fig. 10. Step 364 (Lamping '250, Fig. 9), it will be recalled, is the step that recursively calls the layout routine with the child node's handle, wedge, room bound, and "position".

Thus while the routine of Fig. 9 might represent certain node positions in relative terms within steps 354-362 of Fig. 9, by the time the layout routine of Fig. 9 is called recursively to lay out the next child node, node positions are expressed only in *absolute* terms.

Moreover, prior to step 400 in Fig. 9, the "position" of the current node is not fully developed, even in relative terms. Step 400 calculates the second coordinate of the node's position (the *angle*), then immediately uses the parent node's *absolute* position, together with the distance information and the newly determined angle information, to calculate the *absolute* position of the current node.

Accordingly, while Lamping '632 and '250 might teach obtaining layout data indicating an element's position *relative to a parent* in the space with negative curvature, they do so only as an *intermediate step* in the process of calculating the layout position data for each element. By the time the element's position is written into the node layout data structure, and before the position of the next node is determined even in relative terms, the element's position is expressed only in *absolute* terms.

In particular, there is never a time when the position in layout space of more than one node, whose position has been fully determined, is expressed relative to any other node in the node-link structure being laid out (except perhaps nodes that are children of the root node).

c. Claim 29 calls for the layout position data for *each* of a *plurality* of elements to be laid out, to be represented *only in relative terms* in the data structure after all elements in the plurality have been laid out.

As mentioned, claim 29 calls for, among other things, the step of:

> storing the positions for each element in the plurality in a data structure such that after the positions for all elements in the plurality have been calculated, the position of each element in the plurality is stored in the data structure only relative to an element of the node-link structure other than a root element of the node-link structure. (emphasis added.)

As can be seen, this claim calls for the layout process to identify node positions in relative terms, other than solely as an intermediate step in the calculation of one node's position, by claiming a method of laying out a *plurality* of elements of a node-link structure. The claim calls for the position of *each* element in the plurality to be stored in the data structure only relative to another element of the node-link structure (other than the root). In other words, the positions of *at least two* elements of the node-link structure must be stored in the data structure only in relative terms.

This feature is not described in either Lamping patent. In Lamping '250, in fact, the description is quite clear that position data exists in relative form only as an *intermediate step* in the calculation of the "position" of each node. As soon as the position of a node is known in relative terms, it is immediately converted to absolute terms before the routine begins determining the position of the next node. Never, in Lamping '250, is the position in layout space of *more than one node* represented in the data structure only in relative terms.

Accordingly, claim 29 should be patentable over the Lamping references.

2. Response to Arguments the Examiner *Might* be Making

As mentioned, Appellants submit that the Examiner's new arguments set forth in the Examiner's Answer are so confusing that they fail to state a *prima facie* case of unpatentability.

Nevertheless, in the previous section Appellants analyzed the Lamping patents in detail and pointed out some of the reasons why Lamping cannot anticipate claim 29.

Now, in case the Board discerns a point in the Examiner's argument not answered by the discussion above, Appellants will attempt to answer arguments that they believe the Examiner *might* be making.

The Examiner argues:

> "In response to the arguments in (F), the Examiner notes that, in column 20, lines 27-60 of Lamping '632, Lamping teaches the act in box 362 begins each iteration by obtaining, for the next child node, a position in the hyperbolic plane, a wedge, and a room bound. The act in box 362 uses the distance from box 354 and also uses techniques described in detail in the Layout Application. The room bound can be one-half the distance from the child's parent from box 354 or the radius of a circle in the wedge, whichever is smaller. Lamping '632 thus teaches the relative position of the child node from the parent node in terms of the distance and in terms of the wedge angle and direction in the layout application wherein the act 362 only deals with the child node excluding the parent node. Lamping '632 has taught a wedge of the hyperbolic plane wherein the wedge is available for the descendants of the root node and the wedge is defined by a wedge angle and a direction." (emphasis in Examiner's Answer)

First, to the extent the Examiner is arguing that storage of positions in relative terms is taught by the recursiveness of Lamping's operation in step 362 for obtaining child node positions based on the positions of their parent nodes, Appellants respectfully point out that recursiveness does not imply relativeness. Recursive calls to lay out the positions of the children of a particular node could easily be made by passing down the *absolute* position of the particular node. Such an implementation is in fact likely where, as in Lamping, a tree structure is laid out from the top (root node) down, such that the positions of nodes farther down the tree do not bear on the positions of nodes farther up.

In fact, as explained above, this is how Lamping operates. Lamping obtains the position of each node in relative terms (an absolute direction but a relative distance D from the parent node) only as an intermediate step, and by the time the determination of the position of the next node begins, he has already converted the position of the current node to a fully *absolute* (x,y) position. Never is there a time in Lamping when the positions of each element in a *plurality* of elements are known or stored *relative* to an element of the node-link structure (other than a root element), as called for in Appellants' claim.

Second, to the extent the Examiner is arguing that "the distance from box 354" refers to some distance between two elements in layout space, one of which is a box-shaped feature, Appellants point out that the language of Lamping uses the term "box" to refer to a *step* in a flow chart. "Box 354" refers to *step* 354 in Lamping '632's Fig. 9. Thus "the distance from box 354"

refers to a distance that was calculated in step 354, not a distance between two features in space, one of which is a box-shaped feature.

Third, to the extent the Examiner assumes that a "wedge angle and direction" or a "room bound" in Lamping are expressed relative to a parent, Appellants respectfully point out that a "wedge angle" is the angle-spread of the wedge, and the "room bound" describes a radius of the wedge, from the vertex to an outer arc. (see Lamping '632, col. 20, lines 33-37 for example.) Together they define a pie-shape, and do not by themselves express a particular position in layout space, either relative or absolute. They do not even specify in which direction the shape is oriented, for example. The direction of the wedge is specified by the wedge "direction" term, but even this "direction" is calculated in absolute, not relative terms. Appellants respectfully direct the Board's attention to Lamping '250, col. 23, lines 64-65, where Lamping points out that his directions are defined relative to a predetermined, fixed and absolute "rightward direction, the zero direction from which other directions are measured." They are not expressed relative to a parent.

Finally, the Examiner emphasizes that "<u>Lamping '632 has taught a wedge of the hyperbolic plane wherein the wedge is available for the descendants of the root node and the wedge is defined by a wedge angle and a direction.</u>" (emphasis in original)

Appellants respectfully do not understand how this teaches a step of "storing the positions for each element in the plurality in a data structure such that <u>after the positions for all elements in the plurality</u> have been calculated, <u>the position of each element in the plurality</u> is stored in the data structure <u>only relative</u> to an element of the node-link structure other than a root element of the node-link structure," as called for in Appellants' claim 29.

In Lamping, a "wedge angle" is merely an angle-spread of a wedge, and the "direction" is expressed in *absolute* terms by the time Lamping calls his routine for laying out each next node. Again, therefore, there is never a time in the Lamping patents when the position in layout space of *more than one node* is expressed relative to any other node in the node-link structure, as called for in the claim.

3. Conclusion of Appellants' Argument Regarding Claims 29, 42-44, 30 and 37-41

In light of the above, it is respectfully submitted that the rejections of claims 29, 42-44, 30 and 37-41 should be reversed.

B. Dependent Claim 31 - Examiner's Answer is Perfunctory

In the Appeal Brief, Appellants pointed out reasons why each of the dependent claims 31-36 should be patentable in their own right.

The Examiner's only response to these points appears in the Examiner's Answer at p. 12, lines 9-15, where the following new argument is made:

> "For example, in column 20, lines 20-52, Lamping `632 has taught a wedge of the hyperbolic plane wherein the wedge is available for the descendants of the root node and the wedge is defined by a wedge angle and a direction and the wedge indicates an angular difference between an incoming link to the parent and an outgoing link from the parent to the element, for example, the wedge angle for a node 594 in Fig. 16 indicates the angular difference between the two links to the node 594."

This argument appears to apply only to dependent claim 31, and does not accurately describe how Lamping operates.

In Lamping, the wedge available for laying out descendants of a particular node is an imaginary region of layout space, expressed as a vertex position (the position in layout space of the particular node), a wedge angle and a direction. A "wedge angle" is the angle-spread of the wedge (see Lamping '632, col. 20, lines 33-37 for example), not an angular difference between any incoming link to the particular node and an outgoing link to its child node. It is the "direction" that defines the orientation of the wedge, but even that does not define the direction of any particular descendant node - only the orientation of the imaginary region within which all the descendant nodes will be laid out.

Nor does "the wedge angle for a node 594 in Fig. 16 indicate the angular difference between two links to the node 594," as stated by the Examiner. Lamping '632's Fig. 16 illustrates a *display* image representing a node-link structure after layout has been done, and even after the subsequent mapping to flat display space has been done. The wedge angle used to lay out the children of node 594 was used deep in the recursive layout routine, is long gone before the flat display space image is rendered onto a display, and anyway, never indicated an angular difference between two links to any node.

There is no "wedge angle for a node 594" shown in Fig. 16.

And certainly nothing in Fig. 16 teaches anything about how the position of any node in the image was "represented in the data structure", which is what claim 31 is about.

Thus the cited language of Lamping '632 does not teach that the relative position of any particular element as represented in the data structure include "angle displacement data indicating an angular difference between an incoming link to the parent of the particular element and an outgoing link from the parent to the particular element" as called for in claim 31.

In addition, the Examiner has not cited anything that teaches that such relative position as represented in the data structure include "position displacement data indicating a distance between the particular element and a parent of the particular element", also as called for in claim 31.

The rejection of claim 31 therefore should be reversed.

C. Dependent Claims 32-36 - Examiner Did Not Respond to Appellants' Points

Again, in the Appeal Brief, Appellants pointed out reasons why each of claims 31-36 should be patentable in their own right.

With respect to claims 32-36, it does not appear that the Examiner has submitted any answer.

Accordingly, Appellants submit that the rejections of claims 32-36 should be reversed.

CONCLUSION

In view of the foregoing, Appellants ask that this honorable Board reverse the Examiner's rejections of all the pending claims. In addition, it is submitted that all claims which are the subject of this examination are now allowable, and a notice of intent to issue a patent is respectfully requested.

///

The Commissioner is hereby authorized to charge any fee determined to be due in connection with this communication, or credit any overpayment, to our Deposit Account No. 50-0869 (File No. INXT 1002-1).

Respectfully submitted,

Dated: 01 June 2006

/Warren S. Wolfeld/
Warren S. Wolfeld, Reg. No. 31,454
Attorney for Patent Owner

HAYNES BEFFEL & WOLFELD LLP
P.O. Box 366
751 Kelly Street
Half Moon Bay, CA 94019
Telephone: (650)712.0340
Facsimile: (650)712.0263